

Counting and Enumerating Crossing-free Geometric Graphs

Manuel Wettstein*

April 20, 2016

Abstract

We describe a framework for counting and enumerating various types of crossing-free geometric graphs on a planar point set. The framework generalizes ideas of Alvarez and Seidel, who used them to count triangulations in time $O(2^n n^2)$ where n is the number of points. The main idea is to reduce the problem of counting geometric graphs to counting source-sink paths in a directed acyclic graph.

The following new results will emerge. The number of all crossing-free geometric graphs can be computed in time $O(c^n n^4)$ for some $c < 2.83929$. The number of crossing-free convex partitions can be computed in time $O(2^n n^4)$. The number of crossing-free perfect matchings can be computed in time $O(2^n n^4)$. The number of convex subdivisions can be computed in time $O(2^n n^4)$. The number of crossing-free spanning trees can be computed in time $O(c^n n^4)$ for some $c < 7.04313$. The number of crossing-free spanning cycles can be computed in time $O(c^n n^4)$ for some $c < 5.61804$.

With the same bounds on the running time we can construct data structures which allow fast enumeration of the respective classes. For example, after $O(2^n n^4)$ time of preprocessing we can enumerate the set of all crossing-free perfect matchings using polynomial time per enumerated object. For crossing-free perfect matchings and convex partitions we further obtain enumeration algorithms where the time delay for each (in particular, the first) output is bounded by a polynomial in n .

All described algorithms are comparatively simple, both in terms of their analysis and implementation.

1 Introduction

Let P be a set of n points in the plane. We assume P to be in *general position*, which means that no three points in P are collinear. A *geometric graph* on P is a simple graph with vertex set P , combined with an embedding into the plane where edges are drawn as straight segments between the corresponding endpoints. Two distinct edges are *crossing* if their drawings intersect in their respective relative interiors, otherwise they are *non-crossing*. A geometric graph on P is *crossing-free* if its edges are pairwise non-crossing.

*Department of Computer Science, ETH Zürich, Switzerland. E-mail: mw@inf.ethz.ch

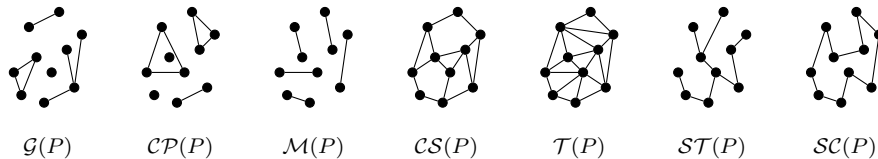


Figure 1: The seven defined classes of crossing-free geometric graphs.

We define the set $\mathcal{G}(P)$ of all crossing-free geometric graphs on P , the set $\mathcal{CP}(P)$ of *crossing-free convex partitions*, the set $\mathcal{M}(P)$ of *crossing-free perfect matchings*, the set $\mathcal{CS}(P)$ of *convex subdivisions*, the set $\mathcal{T}(P)$ of *triangulations*, the set $\mathcal{ST}(P)$ of *crossing-free spanning trees*, and the set $\mathcal{SC}(P)$ of *crossing-free spanning cycles*. Figure 1 shows one representative of each defined class on a fixed point set. Note that by a crossing-free convex partition we mean a partition of P in such a way that the convex hulls of individual parts are pairwise disjoint. Convex subdivisions are subdivisions of the convex hull of P into convex faces with points from P as vertices. Note that every crossing-free convex partition and every convex subdivision is represented uniquely by a crossing-free geometric graph which contains all the edges on the boundaries of individual parts and faces. Finally, a triangulation is an edge-maximal crossing-free geometric graph, which is a special case of a convex subdivision.

Following a common notation, we also define the numbers $\text{pg}(P) := |\mathcal{G}(P)|$, $\text{cp}(P) := |\mathcal{CP}(P)|$, $\text{pm}(P) := |\mathcal{M}(P)|$, $\text{cs}(P) := |\mathcal{CS}(P)|$, $\text{tr}(P) := |\mathcal{T}(P)|$, $\text{st}(P) := |\mathcal{ST}(P)|$, and $\text{sc}(P) := |\mathcal{SC}(P)|$. Some of these numbers have received considerable attention. In particular, exponential upper and lower bounds have been established and gradually improved over the past decades. Instead of listing all successive improvements, which are too numerous to count, we summarize the current state of affairs in Table 1. As a reading example, the top-left entry says that $\text{pg}(P) = O^*(187.53^n)$ holds for all point sets P , see [19] (the $*$ indicates that any subexponential factors are ignored). For an always up-to-date list, which includes bounds for many other types of geometric graphs, we refer the interested reader to Sheffer's webpage [23].

The defined classes of geometric graphs have also been studied from an algorithmic point of view. The problem of enumeration has been solved for $\mathcal{G}(P)$, $\mathcal{T}(P)$ and $\mathcal{ST}(P)$, see [7, 8, 2, 15]. By solved we mean that these sets can be enumerated in such a way that the time delay for each enumerated object is bounded by a polynomial in n .

¹Apply the lower bound for perfect matchings on all subsets of P and then use the binomial theorem.

	$\text{pg}(P)$	$\text{cp}(P)$	$\text{pm}(P)$	$\text{cs}(P)$	$\text{tr}(P)$	$\text{st}(P)$	$\text{sc}(P)$
$\forall P: O^*(c^n)$	187.53 [19]	12.24 [22]	10.05 [22]	187.53 [19]	30.00 [18]	141.07 [12]	54.55 [21]
$\exists P: \Omega^*(c^n)$	41.18 [3]	5.23 [22]	3.00 [11]	8.65 [9]	8.65 [9]	12.52 [13]	4.64 [11]
$\forall P: \Omega^*(c^n)$	11.65 [10]	3.00 [11] ¹	2.00 [11]	2.43 [20]	2.43 [20]	6.75 [10]	1.00

Table 1: Extremal bounds, where cells display the respective exponential bases.

In terms of counting, especially triangulations have been studied extensively [1, 5, 14]. Furthermore, for counting many other types of geometric graphs there already exists a general framework [4], which is based on the onion layer structure of a point set. However, for a long time no counting algorithm was always provably faster than enumerating the set whose size was to be determined. For triangulations this changed with a remarkable paper by Alvarez and Seidel [6], who showed how to compute the number $\text{tr}(P)$ in time $O(2^n n^2)$. This is always exponentially faster than enumeration because $\text{tr}(P) = \Omega^*(2.43^n)$ holds for all sets P , see Table 1. An unrelated approach led to a similar result for the class of all crossing-free geometric graphs. Razen and Welzl [17] showed how to compute $\text{pg}(P)$ in time $O^*(\text{tr}(P))$. Since they proved that $\text{pg}(P) = \Omega^*(2.82^n) \cdot \text{tr}(P)$ holds for all sets P , their algorithm also achieves an exponential speed-up over any procedure that counts by enumerating the whole set $\mathcal{G}(P)$. Furthermore, after the first publication of the present paper [26], Marx and Miltzow discovered a new class of algorithms which make use of the so called cactus layer structure of a triangulation [16]. These algorithms run in time $n^{O(\sqrt{n})}$ and are able to compute the number of triangulations and related structures exactly.

In the remainder of this paper we develop and make use of an abstract framework that generalizes ideas originally used by Alvarez and Seidel [6] for counting triangulations. Loosely speaking, the technique boils down to the following steps. Fix any set of crossing-free geometric graphs whose elements can be decomposed into reasonably small or simple pieces. For instance, every triangulation can be decomposed into a set of interior-disjoint triangles and, similarly, each crossing-free perfect matching can be decomposed into a set of non-intersecting segments. The aim then is to construct a directed acyclic graph Γ with the following properties. Firstly, each edge in Γ is labeled with one of the aforementioned pieces. Secondly, there exist distinguished source and sink vertices in Γ . Thirdly, there is a natural bijection between source-sink paths in Γ and the fixed set of geometric graphs. By this we mean that given any source-sink path in Γ , we can collect all the labels appearing on that path and combine them to obtain the corresponding geometric graph. Clearly, given such a graph Γ for one particular class of geometric graphs, the corresponding counting and enumeration problems can be reduced to counting and enumerating source-sink paths in Γ .

2 Definitions and Results

Let \mathcal{G}_P^* be the set of all crossing-free geometric graphs on all non-empty subsets of P . Elements of \mathcal{G}_P^* are called *units*, and they can be thought of as the simple “pieces” from the informal introduction. For every unit $u \in \mathcal{G}_P^*$ we denote by $\text{pts}(u) \subseteq P$ the set of vertex points of u . That is, if u is a geometric graph on a particular subset P' of P , then $\text{pts}(u) = P'$.

Let us now define a number of useful subsets of \mathcal{G}_P^* .

- Let \mathcal{S}_P be the set of *segments* with both endpoints in P . That is, each $u \in \mathcal{S}_P$ is a geometric graph on exactly two points of P with the edge between them.
- Let \mathcal{CP}_P be the set of *convex parts* with vertex points in P . That is, for each $u \in \mathcal{CP}_P$ the convex hull of $\text{pts}(u)$ does not contain any points of

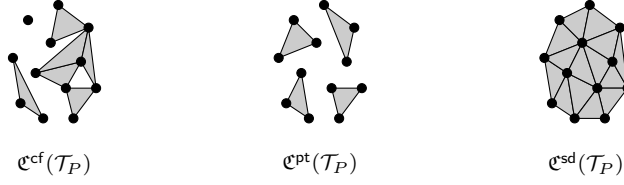


Figure 2: The three defined types of crossing-free combinations of \mathcal{T}_P . The shaded regions represent the shapes of individual elements of \mathcal{T}_P .

$P \setminus \text{pts}(u)$ (in words, interior points are also vertex points of u). Moreover, u contains all the edges along the boundary of the convex hull of the set $\text{pts}(u)$. Observe that all isolated points and segments with endpoints in P are also elements of \mathcal{CP}_P .

- Let \mathcal{CF}_P be the set of *convex faces* with vertex points in P . That is, \mathcal{CF}_P contains all $u \in \mathcal{CP}_P$ with $|\text{pts}(u)| \geq 3$ and no interior vertices. The *shape* of any such u is the bounded and closed region delimited by its edges.
- Let \mathcal{T}_P be the set of *empty triangles* with vertex points in P . That is, \mathcal{T}_P contains all $u \in \mathcal{CF}_P$ with $|\text{pts}(u)| = 3$.

Let us fix a subset \mathcal{U} of \mathcal{G}_P^* . For any units $u_1, u_2 \in \mathcal{U}$, if u_1 contains an edge e_1 and u_2 contains a different edge e_2 such that e_1 and e_2 are crossing, then we also say that u_1 and u_2 are *crossing*. Otherwise, u_1 and u_2 are *non-crossing*. A *combination* of \mathcal{U} is a subset C of \mathcal{U} , and C is *crossing-free* if the elements of C are pairwise non-crossing.

We denote by $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ the set of all crossing-free combinations of \mathcal{U} . Furthermore, a combination $C \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$ is called a *partition* of P if the sets $\text{pts}(u)$ of all $u \in C$ are pairwise disjoint and if their union is equal to P . We denote by $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ the set of all such partitions of P . Assuming $\mathcal{U} \subseteq \mathcal{CF}_P$, a combination $C \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$ is called a *subdivision* of the convex hull of P if the shapes of all $u \in C$ are pairwise interior-disjoint and if their union is equal to the convex hull of P . We denote by $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ the set of all such subdivisions of the convex hull of P . Figure 2 shows one representative of each type of crossing-free combinations for the special case $\mathcal{U} = \mathcal{T}_P$.

From now on we will no longer consider the sets $\mathcal{G}(P)$, $\mathcal{CP}(P)$, and so on, as defined in the introduction. Instead, we will talk about crossing-free combinations of specific sets of units. For example, since there is an obvious bijection between the sets $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ and $\mathcal{G}(P)$, any counting or enumeration algorithm for one set can be adapted easily for the other. Similarly, there are bijections between the sets $\mathfrak{C}^{\text{pt}}(\mathcal{CP}_P)$ and $\mathcal{CP}(P)$, $\mathfrak{C}^{\text{pt}}(\mathcal{S}_P)$ and $\mathcal{M}(P)$, $\mathfrak{C}^{\text{sd}}(\mathcal{CF}_P)$ and $\mathcal{CS}(P)$, as well as $\mathfrak{C}^{\text{sd}}(\mathcal{T}_P)$ and $\mathcal{T}(P)$.

In the same spirit, we define the sets $\mathfrak{C}_P^{\text{st}}, \mathfrak{C}_P^{\text{sc}} \subseteq \mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ of all crossing-free combinations of \mathcal{S}_P whose segments form spanning trees and spanning cycles on P , respectively.

Definition 1. Let $\mathcal{U} \subseteq \mathcal{G}_P^*$. A *combination graph* (over \mathcal{U}) is a directed and acyclic multigraph Γ with two distinguished vertices \perp and \top , called the source and sink of Γ , respectively. All edges in Γ , except for those ending in \top , are

labeled with an element of \mathcal{U} . Moreover, the sink \top has no outgoing edges. The size of Γ is the number of vertices and edges in Γ .

For any combination graph Γ and any set \mathfrak{C} of combinations, we say that Γ *represents* \mathfrak{C} if there is a bijection between the set of directed \perp - \top paths in Γ and the set \mathfrak{C} in the following sense. Taking any \perp - \top path in Γ and building the set of labels on that path yields the corresponding combination in \mathfrak{C} .

The following are comparatively simple applications of an abstract framework developed in Section 3. The corresponding proofs can be found in Section 4. Many more, in some cases obvious, applications are possible.

Theorem 1 (All geometric graphs). *There exists a combination graph over \mathcal{S}_P of size $O(c^n n^3)$ with $c < 2.83929$ that represents $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$.*

Theorem 2 (Convex partitions). *There exists a combination graph over \mathcal{CP}_P of size $O(2^n n^3)$ that represents $\mathfrak{C}^{\text{pt}}(\mathcal{CP}_P)$.*

Theorem 3 (Perfect matchings). *There exists a combination graph over \mathcal{S}_P of size $O(2^n n^3)$ that represents $\mathfrak{C}^{\text{pt}}(\mathcal{S}_P)$.*

Theorem 4 (Convex subdivisions). *There exists a combination graph over \mathcal{CF}_P of size $O(2^n n^3)$ that represents $\mathfrak{C}^{\text{sd}}(\mathcal{CF}_P)$.*

Theorem 5 (Triangulations; c.f. Theorem 3 in [6]). *There exists a combination graph over \mathcal{T}_P of size $O(2^n n^3)$ that represents $\mathfrak{C}^{\text{sd}}(\mathcal{T}_P)$.*

Within our framework we can explain similar results for spanning trees and spanning cycles. However, these two classes are substantially harder to deal with. Section 7 is devoted to the corresponding proofs.

Theorem 6 (Spanning trees). *There exists a combination graph over \mathcal{S}_P of size $O(c^n n^3)$ with $c < 7.04313$ that represents $\mathfrak{C}_P^{\text{st}}$.*

Theorem 7 (Spanning cycles). *There exists a combination graph over \mathcal{S}_P of size $O(c^n n^3)$ with $c < 5.61804$ that represents $\mathfrak{C}_P^{\text{sc}}$.*

To get a bound on the running time for computing an explicit representation of any one of the combination graphs Γ in the above theorems, it suffices to add another factor n to the bound on the size of Γ . See Section 5 for details.

Given such a representation of Γ , the corresponding counting problem can be solved in time linear in the size of Γ by counting directed \perp - \top paths using standard graph algorithms. After removing all dead ends in Γ , which is also possible in time linear in the size of Γ , enumeration of the corresponding set \mathfrak{C} requires time at most linear in the length of the longest \perp - \top path per enumerated object. We will abstain from describing the required algorithms in detail, and instead refer to [6] for an example.

Observe that the exponential bases in Theorems 1 to 5 are not larger than the exponential bases of the corresponding lower bounds given in the third row of Table 1. As a corollary we therefore get enumeration algorithms for the sets $\mathcal{G}(P)$, $\mathcal{CP}(P)$, $\mathcal{M}(P)$, $\mathcal{CS}(P)$ and $\mathcal{T}(P)$ whose overall running times are bounded by the length of the output times a polynomial in n . For $\mathcal{CP}(P)$ and $\mathcal{M}(P)$, a small adaptation, which is described in Section 6, results in enumeration algorithms with polynomial delay for each (in particular, the first) output.

Theorem 8. *The sets $\mathcal{CP}(P)$ and $\mathcal{M}(P)$ can be enumerated such that the time delay for any output is bounded by a polynomial in n .*

With the exception of $\mathcal{M}(P)$, the lower bounds are even strictly larger, which means that our algorithms compute the numbers $\text{pg}(P)$, $\text{cp}(P)$, $\text{cs}(P)$ and $\text{tr}(P)$ with exponential speed-up over any procedures that count by enumerating the respective sets. For spanning trees it might as well be that the constant c in Theorem 6 is smaller than 6.75, but we were unable to prove it. For spanning cycles we cannot always hope for such an exponential speed-up because for a set P of n points in convex position we have $\text{sc}(P) = 1$.

3 The Abstract Framework

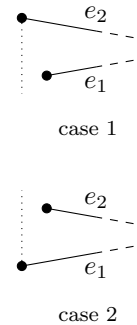
As before, let P be a set of n points in general position. Furthermore, assume that no two points share the same x -coordinate, which means the points can be ordered as p_1, \dots, p_n from left to right in a unique way. If a point p_i is to the left of another point p_j , that is, if $i \leq j$, then we write $p_i \preceq p_j$. Recall that $\text{pts}(u) \subseteq P$ is the set of vertex points in any $u \in \mathcal{G}_P^*$. We define $\text{lft}(u) := \min_{\preceq}(\text{pts}(u))$ and $\text{rgt}(u) := \max_{\preceq}(\text{pts}(u))$, the *left-most* and *right-most point* of u , respectively. For any $u_1, u_2 \in \mathcal{G}_P^*$, if $\text{rgt}(u_1) \preceq \text{lft}(u_2)$ holds then we say that u_1 is *to the left of* u_2 and we also write $u_1 \preceq u_2$.

For each u we define $\text{low}(u) \subseteq P$ and $\text{upp}(u) \subseteq P$, the *lower* and *upper shadow* of u , respectively. The set $\text{low}(u)$ contains all points in P from which a vertical ray shooting upwards intersects the relative interior of some segment of u . The set $\text{upp}(u)$ is defined in an analogous way. Whenever we have either $\text{pts}(u_1) \cap \text{low}(u_2) \neq \emptyset$ or $\text{upp}(u_1) \cap \text{pts}(u_2) \neq \emptyset$ for any u_1, u_2 , then we say that u_2 *depends on* u_1 and we write $u_1 \sqsubset u_2$. The following lemma aims at making this cryptic definition more intuitive.

Here, and only here, by a *point on* u we mean either a point in $\text{pts}(u)$ or a point in the relative interior of some edge of u .

Lemma 1. *Let $u_1, u_2 \in \mathcal{G}_P^*$ be arbitrary. Then, u_2 depends on u_1 if and only if there exists a point on u_2 directly (that is, same x -coordinate) and strictly above a point on u_1 . In particular, if u_1 and u_2 are crossing then they are mutually dependent, that is, $u_1 \sqsubset u_2$ and $u_2 \sqsubset u_1$.*

Proof. The “only if” is immediate by definition of $u_1 \sqsubset u_2$. For the “if”, let us fix two points on u_1 and u_2 , respectively, the one on u_2 directly and strictly above the one on u_1 . If either of those points is contained in $\text{pts}(u_1)$ or $\text{pts}(u_2)$, respectively, the conclusion $u_1 \sqsubset u_2$ is again immediate. Otherwise, let e_1 and e_2 be the edges of u_1 and u_2 that contain the two respective fixed points in their relative interiors. Without loss of generality we assume that e_1 and e_2 diverge and thus do not intersect towards the left, as illustrated on the right. In case 1, the left endpoint of e_1 is below e_2 , which means $\text{pts}(u_1) \cap \text{low}(u_2) \neq \emptyset$ and hence $u_1 \sqsubset u_2$. In case 2, the left endpoint of e_2 is above e_1 , which means $\text{upp}(u_1) \cap \text{pts}(u_2) \neq \emptyset$ and hence again $u_1 \sqsubset u_2$. \square



To ease notation throughout, we use $\text{pts}(C)$ and $\text{low}(C)$ to denote the sets $\bigcup_{u \in C} \text{pts}(u)$ and $\bigcup_{u \in C} \text{low}(u)$, respectively, where C is any combination.

Besides giving an intuition for the dependence relation, the above lemma turns out to be absolutely crucial for everything that follows. It suggests a safe and practical way of adding a new segment $u \in \mathcal{S}_P$, say, to a crossing-free combination C of \mathcal{S}_P . Safe means that we do not introduce any crossings, that is, $C \cup \{u\}$ is itself crossing-free. Practical means that we may, to a great extent, remain ignorant of the exact composition of C . Indeed, as long as we know the sets $\text{pts}(C)$ and $\text{low}(C)$, and provided that we choose u such that $\text{pts}(u) \cap \text{low}(C) = \emptyset$ and $\text{upp}(u) \cap \text{pts}(C) = \emptyset$, then no element of C can possibly depend on u and hence, by Lemma 1, $C \cup \{u\}$ is crossing-free.

By extension, this suggests a way of constructing a combination C by adding all elements in a succession where earlier occurrences do not depend on later occurrences. An apparent disadvantage is that this will not work for every conceivable subset \mathcal{U} of \mathcal{G}_P^* and every crossing-free combination C of \mathcal{U} . Most importantly, for it to work, there must be no circular dependencies among elements of C . In the following we formalize this requirement.

Let $\mathcal{U} \subseteq \mathcal{G}_P^*$ and let $C \subseteq \mathcal{U}$ be arbitrary. An element u of C is *extreme (in C)* if $u \not\preceq u'$ holds for all other elements u' of C . If it exists, the *right-most extreme element* in C is the unique extreme element u in C which satisfies $u' \preceq u$ for all other extreme elements u' in C .

Definition 2. Let $\mathcal{U} \subseteq \mathcal{G}_P^*$ and let \mathfrak{C} be a set of combinations of \mathcal{U} . We call \mathfrak{C} *serializable* if it is non-empty and if every non-empty $C \in \mathfrak{C}$ contains a right-most extreme element, which we then denote by $\text{rex}(C)$. Additionally, $C \setminus \{\text{rex}(C)\}$ must itself be an element of \mathfrak{C} .

Let \mathfrak{C} be a serializable set of combinations of \mathcal{U} , and let $C, C' \in \mathfrak{C}$. We will often write $C \xrightarrow{u} C'$, which stands for $C = C' \setminus \{u\}$ and $u = \text{rex}(C')$. Observe that \mathfrak{C} naturally induces a directed and acyclic graph (actually, a tree) with vertex set \mathfrak{C} and edges with labels from the set \mathcal{U} . Indeed, whenever $C \xrightarrow{u} C'$ holds we simply add an edge from vertex C to vertex C' with label u . A combination graph over \mathcal{U} that represents an arbitrary subset of \mathfrak{C} is obtained by defining $\perp := \emptyset$ and by adding appropriate unlabeled edges which end in an additional vertex \top . However, the resulting combination graph is too large as its size is $\Theta(|\mathfrak{C}|)$. In the following we show how to compress it without losing too many of its nice properties.

Definition 3. Let \mathfrak{C} be serializable. An equivalence relation \sim on \mathfrak{C} is called *coherent* if $C_1 \sim C_2$ implies the following. If $C_1 \xrightarrow{u} C'_1$ holds then also $C_2 \xrightarrow{u} C'_2$ for some $C'_2 \sim C'_1$.

Intuitively, to make our combination graph smaller we would like to merge two vertices C_1 and C_2 . This makes sense only if the subtrees rooted at C_1 and C_2 are identical when looking at edge labels. As will be proved later, if $C_1 \sim C_2$ holds, coherency of \sim enforces precisely what we want.

In the remainder of the paper we will always deal with a serializable set \mathfrak{C} of combinations of some $\mathcal{U} \subseteq \mathcal{G}_P^*$, and an equivalence relation \sim on \mathfrak{C} . For any $C \in \mathfrak{C}$ we then define the equivalence class $[C] := \{C' \in \mathfrak{C} : C' \sim C\}$, where the relation \sim will be obvious from the context. We also define the set $(\mathfrak{C}/\sim) := \{[C] : C \in \mathfrak{C}\}$ of all equivalence classes.

Definition 4. A combination problem (on P) is a tuple $(\mathcal{U}, \mathfrak{C}, \sim, \mathfrak{T})$ where $\mathcal{U} \subseteq \mathcal{G}_P^*$, \mathfrak{C} is a serializable set of combinations of \mathcal{U} , \sim is a coherent equivalence relation on \mathfrak{C} , and \mathfrak{T} is a subset of (\mathfrak{C}/\sim) .

As will be described below, the set \mathfrak{T} allows us to specify the subset of \mathfrak{C} we are actually interested in.

Every combination problem $(\mathcal{U}, \mathfrak{C}, \sim, \mathfrak{T})$ induces a corresponding combination graph $\Gamma = \Gamma(\mathcal{U}, \mathfrak{C}, \sim, \mathfrak{T})$ over \mathcal{U} as follows. The vertices of Γ are all equivalence classes in (\mathfrak{C}/\sim) plus one extra vertex denoted by \perp . The vertex $[\emptyset]$ is also referred to by \perp . Existence of \perp follows from $\emptyset \in \mathfrak{C}$, an easy consequence of serializability of \mathfrak{C} . Whenever $C \xrightarrow{u} C'$ holds, then we add an edge from vertex $[C]$ to vertex $[C']$ with label u . We do not, however, add the same labeled edge more than once. Lastly, for every vertex $[C]$ in \mathfrak{T} we add an unlabeled edge which starts in $[C]$ and ends in \perp . Observe that Γ does not contain any directed cycles because given any such cycle, it would be possible to construct an infinite sequence $(C \xrightarrow{u} C', C' \xrightarrow{u} C'', \dots)$, which cannot exist.

Recall that $C \xrightarrow{u} C'$ stands for $C = C' \setminus \{u\}$ and $u = \text{rex}(C')$. Additionally, we write $[C] \xrightarrow{u} [C']$ if there exists an edge in Γ from vertex $[C]$ to vertex $[C']$ with label u . The following observations are straight-forward consequences of coherency of \sim and of the way Γ is constructed.

Observation 1. *The combination graph $\Gamma = \Gamma(\mathcal{U}, \mathfrak{C}, \sim, \mathfrak{T})$, defined as above, is*

- *Complete: if $C \xrightarrow{u} C'$, then $[C] \xrightarrow{u} [C']$,*
- *Sound: if $[C] \xrightarrow{u} [C']$, then $C \xrightarrow{u} C''$ for some $C'' \sim C'$,*
- *Deterministic: if $[C] \xrightarrow{u} [C']$ and $[C] \xrightarrow{u} [C'']$, then $[C'] = [C'']$.*

By induction it can now be shown that for every vertex $[C]$ there is a natural bijection from $[C]$ to the set of directed \perp - $[C]$ paths in Γ , which implies the following lemma.

Lemma 2. *Let $(\mathcal{U}, \mathfrak{C}, \sim, \mathfrak{T})$ be a combination problem and let Γ be the corresponding combination graph. Then, Γ represents $\bigcup \mathfrak{T}$ and the size of Γ is at most $O(|(\mathfrak{C}/\sim)| \cdot |\mathcal{U}|)$.*

Proof. The upper bound on the size of Γ holds because it has exactly $|(\mathfrak{C}/\sim)| + 1$ vertices and, since it is deterministic (as in Observation 1) and has no duplicate labeled edges, each vertex has at most $|\mathcal{U}| + 1$ outgoing edges. It only remains to show that Γ represents $\bigcup \mathfrak{T}$.

For each directed path R in Γ let C_R denote the set of all labels on R , that is, C_R is the combination of \mathcal{U} corresponding to the path R . Moreover, for any $C \in \mathfrak{C}$, the *canonical order* of C is the unique sequence over all elements in C which is obtained by successively removing right-most extreme elements from C and then putting the removed elements in reverse order.

For every vertex $[C]$ in Γ we now claim the following. Firstly, there exists a directed \perp - $[C]$ path R in Γ with $C_R = C$, irrespective of the choice of the representative C . Secondly, for every directed \perp - $[C]$ path R we have that $C_R \in [C]$. Thirdly, the labels on any directed \perp - $[C]$ path R with $C_R = C$ appear in canonical order of C . The proof of these claims is by induction over an arbitrary topological ordering of the vertices in Γ . After that, the lemma follows by combining the claims with the fact that Γ is deterministic.

For the first part of the claim, if $C = \emptyset$, then $[C] = \perp$ and the \perp - \perp path without edges works. Otherwise, if $C \neq \emptyset$, then we know that $C' \xrightarrow{u} C$ for some $C' \in \mathfrak{C}$ and u . By completeness, there is an edge $[C'] \xrightarrow{u} [C]$ in Γ and a \perp - $[C']$ path

R with $C_R = C$ can be constructed from the $\perp\text{-}[C']$ path R' with $C_{R'} = C'$, which exists by induction.

For the second part of the claim, let R be a directed $\perp\text{-}[C]$ path in Γ . If R is of length zero then $[C] = \perp$ and $C_R = \emptyset \in [C]$. Otherwise, let R' be the $\perp\text{-}[C']$ path that is obtained by removing the last edge $[C'] \xrightarrow{u} [C]$ from R . By induction, and without loss of generality, $C_{R'} = C'$. By soundness, we have $C' \xrightarrow{u} C''$ for some $C'' \sim C$ and hence also $C_R = C'' \in [C]$.

For the third part of the claim, simply observe that in the previous paragraph, the last label on the path R with $C_R = C''$ is the right-most extreme element in C'' . Hence, this last claim also follows by induction. \square

4 Three Simple Applications

We present three generic kinds of combination problems. They directly correspond to the three types of crossing-free combinations depicted earlier in Figure 2.

4.1 All Crossing-free Combinations

The main aim of this subsection is to give a proof of Theorem 1. Still, some of the following insights are fairly general and can be used in many different settings.

Let us fix a set of units $\mathcal{U} \subseteq \mathcal{G}_P^*$. We make the explicit assumption that $\mathfrak{C}^{\text{cf}}(\mathcal{U})$, the set of crossing-free combinations of \mathcal{U} , is serializable. Then, as follows, a combination C in $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ can be described by a coloring of the point set P with three colors \circ, \otimes, \bullet and a special marking, e.g., \otimes , on one of the points. A point $p \in P$ is given the color \bullet if $p \in \text{low}(C)$, it is given the color \otimes if $p \in \text{pts}(C) \setminus \text{low}(C)$, and it is given the color \circ in all other cases. The special marking is put on the left-most point of $\text{rex}(C)$. Figure 3 shows, for the special case $\mathcal{U} = \mathcal{S}_P$, that different crossing-free combinations can have identical such descriptions. Whenever that is the case, we consider two combinations equivalent. More formally, we put $C \sim C'$ if and only if²

- $\text{low}(C) = \text{low}(C')$,
- $\text{pts}(C) \setminus \text{low}(C) = \text{pts}(C') \setminus \text{low}(C')$,
- $\text{lft}(\text{rex}(C)) = \text{lft}(\text{rex}(C'))$.

²Of course, the third condition only makes sense if C and C' are non-empty.

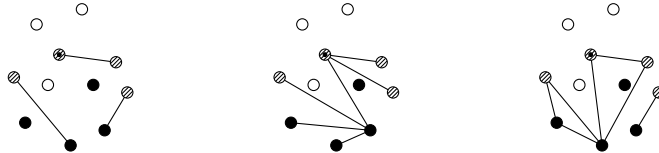


Figure 3: Three elements of $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ that are considered equivalent.

The goal now is to prove that \sim is coherent, as in Definition 3. Unfortunately, this endeavor is doomed to fail because for some contrived choices of \mathcal{U} we can have $C \xrightarrow{u} C'$ and, at the same time, $C \sim C'$. In the language of combination graphs this means that we would have to introduce loops, which leads to a potentially infinite number of source-sink paths. To avoid this problem, we require for any $C, C' \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$ and any $u \in \mathcal{U}$ that $C \xrightarrow{u} C'$ implies $C \not\sim C'$. If this additional requirement is met, then we say that $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ is *progressive*.

Lemma 3. *Let $\mathcal{U} \subseteq \mathcal{G}_P^*$ with $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ both serializable and progressive. Then, the equivalence relation \sim on $\mathfrak{C}^{\text{cf}}(\mathcal{U})$, as defined above, is coherent.*

Proof. Let $C_1, C_2 \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$ be non-empty (otherwise, the proof is trivial) with $C_1 \sim C_2$ and assume that $C_1 \xrightarrow{u} C'_1$ holds for $C'_1 \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$ and $u \in \mathcal{U}$. Consider $C'_2 := C_2 \cup \{u\}$. We show $C'_2 \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$, $C'_1 \sim C'_2$ and $C_2 \xrightarrow{u} C'_2$, which implies the lemma.

By assumption we have $\text{low}(C_1) = \text{low}(C_2)$ and $\text{pts}(C_1) \setminus \text{low}(C_1) = \text{pts}(C_2) \setminus \text{low}(C_2)$. From this and from the fact that u is extreme in C'_1 , we first derive that u is also extreme in C'_2 . So, assume the opposite, that is, either $\text{pts}(u) \cap \text{low}(C_2) \neq \emptyset$ or $\text{upp}(u) \cap \text{pts}(C_2) \neq \emptyset$ holds. In the first case, we get $\text{pts}(u) \cap \text{low}(C_1) = \text{pts}(u) \cap \text{low}(C_2) \neq \emptyset$, contradicting that u is extreme in C'_1 . In the second case, we get at least one of $\text{upp}(u) \cap \text{pts}(C_1) \supseteq \text{upp}(u) \cap (\text{pts}(C_1) \setminus \text{low}(C_1)) = \text{upp}(u) \cap (\text{pts}(C_2) \setminus \text{low}(C_2)) \neq \emptyset$ and $\text{upp}(u) \cap \text{low}(C_1) = \text{upp}(u) \cap \text{low}(C_2) \supseteq \text{upp}(u) \cap (\text{pts}(C_2) \cap \text{low}(C_2)) \neq \emptyset$. Both possibilities again lead to u not being extreme in C'_1 . We conclude that, indeed, u is extreme in C'_2 .

Since u is extreme in C'_2 , no other element of C'_2 depends on u . By Lemma 1, since C_2 is crossing-free, also C'_2 must be crossing-free. Hence, $C'_2 \in \mathfrak{C}^{\text{cf}}(\mathcal{U})$.

Next, by definition of C'_2 it is easily seen that $\text{low}(C'_1) = \text{low}(C'_2)$ and $\text{pts}(C'_1) \setminus \text{low}(C'_1) = \text{pts}(C'_2) \setminus \text{low}(C'_2)$ both hold. To prove also $\text{lft}(\text{rex}(C'_1)) = \text{lft}(\text{rex}(C'_2))$, and consequently $C'_1 \sim C'_2$, it suffices to show $\text{rex}(C'_2) = u$. So, for $u' := \text{rex}(C'_2)$ and assuming that $u' \neq u$, we get $u \preceq u'$ and thus also $\text{rex}(C_2) = u'$. Invoking $C_1 \sim C_2$ yields $u \preceq \text{rex}(C_1)$, which contradicts the assumption $u = \text{rex}(C'_1)$.

It remains to prove $C_2 \xrightarrow{u} C'_2$. The crucial observation here is that $u = \text{rex}(C'_2)$ and $C_2 \cup \{u\} = C'_2$ alone are not sufficient. What we need instead is $C_2 = C'_2 \setminus \{u\}$, that is, u must not be contained in C_2 . However, assuming that u is contained in C_2 , we easily derive $C_1 \sim C_2 = C'_2 \sim C'_1$, which, when combined with $C_1 \xrightarrow{u} C'_1$, contradicts the fact that $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ is progressive. \square

We define $\mathfrak{T} := (\mathfrak{C}^{\text{cf}}(\mathcal{U})/\sim)$. With Lemma 3 we see that $(\mathcal{U}, \mathfrak{C}^{\text{cf}}(\mathcal{U}), \sim, \mathfrak{T})$ is a combination problem as long as $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ is serializable and progressive. The corresponding combination graph over \mathcal{U} is referred to by $\Gamma_{\mathcal{U}}^{\text{cf}}$.

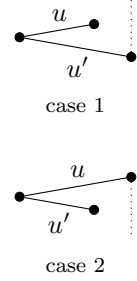
Corollary 1. *If $\mathcal{U} \subseteq \mathcal{G}_P^*$ is such that $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ is both serializable and progressive, then $\Gamma_{\mathcal{U}}^{\text{cf}}$ represents $\mathfrak{C}^{\text{cf}}(\mathcal{U})$ and the size of $\Gamma_{\mathcal{U}}^{\text{cf}}$ is at most $O(|(\mathfrak{C}^{\text{cf}}(\mathcal{U})/\sim)| \cdot |\mathcal{U}|)$.*

Theorem 1 follows by invoking Corollary 1 with $\mathcal{U} = \mathcal{S}_P$ and by making use of the next two lemmas. The first one essentially shows that $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ is a well-behaved set. The second one gives us a better upper bound on the size of $\Gamma_{\mathcal{S}_P}^{\text{cf}}$. Observe, however, that a bound of $O(3^n n^3)$ is immediate because we can encode equivalence classes with 3 colors and with one marking, and because of $|\mathcal{S}_P| = O(n^2)$.

Lemma 4. For any point set P , $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ is both serializable and progressive.

Proof (serializable). Let $C \in \mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ be arbitrary but non-empty. We prove serializability by exhibiting a right-most extreme element u in C . For this, consider the relative interiors of all segments in C . These segments without endpoints are convex and pairwise non-intersecting. For a set of convex and non-intersecting shapes in the plane it is well known that at least one of them can be translated in the y -direction to infinity in a continuous motion, without intersecting any other segment in the process [24]. Any segment with this property corresponds to an extreme element in C . The extreme elements in C now can be ordered from left to right according to their relative positions when projected orthogonally onto the x -axis. The segment u on the far right is a right-most extreme element in C . \square

Proof (progressive). For the sake of contradiction, assume that we have $C' \xrightarrow{u} C$ and $C \sim C'$ for some $C, C' \in \mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ and $u \in \mathcal{S}_P$. Let $u' := \text{rex}(C')$ (note that the case where C' is empty is trivial). We have $\text{lft}(u) = \text{lft}(u')$ and we have to distinguish the two cases illustrated on the right. In case 1, we see that $\text{rgt}(u) \in \text{pts}(C) \setminus \text{low}(C)$ but also $\text{rgt}(u) \notin \text{pts}(C') \setminus \text{low}(C')$ since otherwise u' would not be extreme in C' . In case 2, we see that $\text{rgt}(u') \in \text{low}(C)$ but also $\text{rgt}(u') \notin \text{low}(C')$ since otherwise u' would not be extreme in C' . Hence, in both cases we get a contradiction to the assumption $C \sim C'$. \square



Lemma 5. For any set P of n points, the relation \sim partitions $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ into at most $O(\alpha^n n)$ equivalence classes, that is, $|(\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)/\sim)| = O(\alpha^n n)$, where $\alpha \lesssim 2.83929$.³

The proof of Lemma 5 is a bit tedious. We only sketch the main idea here, and postpone a more careful analysis to Section 8.

Note that for any three consecutive points p_i, p_{i+1}, p_{i+2} in P , the point p_{i+1} is either below or above the straight line through p_i and p_{i+2} , as depicted in Figure 4. In both cases we can show that at least one of the $3^3 = 27$ different ways of assigning colors to p_i, p_{i+1}, p_{i+2} does not describe an actual element of $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$. A bound of $O(26^{n/3} n) = O(2.963^n n)$ on the size of $(\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)/\sim)$ then follows after partitioning P into $n/3$ consecutive triples.

In the first case, if we assign colors \bullet, \circ, \bullet , there must be two distinct segments which pass over the points p_i and p_{i+2} , respectively, and which pass under p_{i+1} . Clearly, any two such segments are crossing. In the second case, if we assign colors \circ, \bullet, \circ , there must be a segment that passes over p_{i+1} and under p_i and p_{i+2} . Clearly, only a bent “segment” can achieve that.

³We write $\alpha \lesssim \beta$ if a parameter α is approximately equal to β and also strictly smaller.



Figure 4: Three consecutive points can always be colored in a way that does not correspond to an element of $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$.

Note also that if the combination graph $\Gamma_{\mathcal{G}_P}^{\text{cf}}$ is constructed bottom-up, as will be explained in more detail in Section 5, then all these impossible colorings are avoided automatically.

4.2 Crossing-free Partitions

Let us again fix a set of units $\mathcal{U} \subseteq \mathcal{G}_P^*$. We define the set $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ of all crossing-free combinations C of \mathcal{U} for which the sets $\text{pts}(u)$ of all $u \in C$ are pairwise disjoint and for which $\text{low}(C) \subseteq \text{pts}(C)$ holds. Figure 5 depicts three of these combinations for the special case $\mathcal{U} = \mathcal{CP}_P$. Observe that $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ and $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ are different sets, and also observe that we have in fact $\mathfrak{C}^{\text{pt}}(\mathcal{U}) \subseteq \mathfrak{C}^{\text{pt}}(\mathcal{U})$.

Assume that $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ is serializable. Similar to the previous subsection, we use two colors \circ, \otimes and a special marking \bullet on the points in P to describe an element C of $\mathfrak{C}^{\text{pt}}(\mathcal{U})$. A point p receives the color \otimes if $p \in \text{pts}(C)$, and \circ otherwise. The marking is again put on the left-most point of $\text{rex}(C)$. If two combinations have identical such descriptions, we consider them equivalent. Formally, we put $C \sim C'$ if and only if

- $\text{pts}(C) = \text{pts}(C')$,
- $\text{lft}(\text{rex}(C)) = \text{lft}(\text{rex}(C'))$.

One peculiarity in Figure 5 is that some points have been given the color \otimes even though there are no incident segments. This is because the set \mathcal{CP}_P also contains all isolated points, that is, all $u \in \mathcal{G}_P^*$ with $|\text{pts}(u)| = 1$.

Also observe that $C \xrightarrow{u} C'$ implies $C \not\sim C'$ because $\text{pts}(C)$ is always a proper subset of $\text{pts}(C')$. Consequently, an explicit notion of “progressive”, as in the previous subsection, is not needed here.

Lemma 6. *Let $\mathcal{U} \subseteq \mathcal{G}_P^*$ with $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ serializable. Then, the equivalence relation \sim on $\mathfrak{C}^{\text{pt}}(\mathcal{U})$, as defined above, is coherent.*

Proof. Let $C_1, C_2 \in \mathfrak{C}^{\text{pt}}(\mathcal{U})$ be non-empty (otherwise, the proof is trivial) with $C_1 \sim C_2$ and assume that $C_1 \xrightarrow{u} C'_1$ holds for $C'_1 \in \mathfrak{C}^{\text{pt}}(\mathcal{U})$ and $u \in \mathcal{U}$. Consider $C'_2 := C_2 \cup \{u\}$. We show $C'_2 \in \mathfrak{C}^{\text{pt}}(\mathcal{U})$, $C'_1 \sim C'_2$ and $C_2 \xrightarrow{u} C'_2$, which implies the lemma.

First, observe that since u is extreme in C'_1 , it is also extreme in C'_2 . Indeed, $\text{pts}(u) \cap \text{low}(C_2) \subseteq \text{pts}(u) \cap \text{pts}(C_2) = \text{pts}(u) \cap \text{pts}(C_1) = \emptyset$ and $\text{upp}(u) \cap \text{pts}(C_2) = \text{upp}(u) \cap \text{pts}(C_1) = \emptyset$ can be derived immediately. By Lemma 1, C'_2 is crossing-free. Moreover, deriving $\text{low}(C'_2) = \text{low}(C_2) \cup \text{low}(u) \subseteq \text{pts}(C_2) \cup \text{low}(C'_1) \subseteq \text{pts}(C'_2) \cup \text{pts}(C'_1) = \text{pts}(C'_2)$ proves that, indeed, $C'_2 \in \mathfrak{C}^{\text{pt}}(\mathcal{U})$. By making use of the marking in the same way as in the proof of Lemma 3 we get $\text{rex}(C'_2) = u$, and it follows that $C'_1 \sim C'_2$ and $C_2 \xrightarrow{u} C'_2$. \square

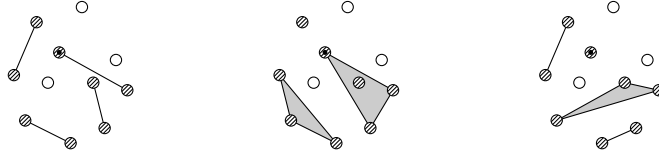


Figure 5: Three elements of $\mathfrak{C}^{\text{pt}}(\mathcal{CP}_P)$ that are considered equivalent.

We define the set $\mathfrak{T} \subseteq (\mathfrak{C}^{\text{pt}}(\mathcal{U})/\sim)$ which contains all equivalence classes $[C]$ for which $\text{pts}(C) = P$ holds. Observe that then $\bigcup \mathfrak{T} = \mathfrak{C}^{\text{pt}}(\mathcal{U})$. From Lemma 6 it follows that $(\mathcal{U}, \mathfrak{C}^{\text{pt}}(\mathcal{U}), \sim, \mathfrak{T})$ is a combination problem provided that $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ is serializable. We denote by $\Gamma_{\mathcal{U}}^{\text{pt}}$ the corresponding combination graph.

Corollary 2. *If $\mathcal{U} \subseteq \mathcal{G}_P^*$ is such that $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ is serializable, then $\Gamma_{\mathcal{U}}^{\text{pt}}$ represents $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ and the size of $\Gamma_{\mathcal{U}}^{\text{pt}}$ is at most $O(|(\mathfrak{C}^{\text{pt}}(\mathcal{U})/\sim)| \cdot |\mathcal{U}|)$.*

Theorems 2 and 3 now follow from Corollary 2 and the following two lemmas.

Lemma 7. *For any point set P and any subset \mathcal{U} of \mathcal{CP}_P , it holds that $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ is serializable.*

Proof. The proof for the existence of right-most extreme elements is analogous to the first part of the proof of Lemma 4. However, proving that $C' := C \setminus \{u\}$, where $u = \text{rex}(C)$, is also an element of $\mathfrak{C}^{\text{pt}}(\mathcal{U})$ is not completely trivial since we have to verify that $\text{low}(C') \subseteq \text{pts}(C')$ holds.

For the sake of contradiction, let us assume that there exists a point $p \in \text{low}(C') \setminus \text{pts}(C')$. From $C' \subseteq C$ we get $\text{low}(C') \subseteq \text{low}(C)$ and hence also $p \in \text{low}(C)$. We now use $\text{low}(C) \subseteq \text{pts}(C)$, which holds by definition of $\mathfrak{C}^{\text{pt}}(\mathcal{U})$, to obtain $p \in \text{pts}(C)$. By combining this with the assumption $p \notin \text{pts}(C')$ we obtain $p \in \text{pts}(u)$ because $\text{pts}(C) \setminus \text{pts}(C') = \text{pts}(u)$. It follows that $\text{low}(C') \cap \text{pts}(u)$ is non-empty since it contains at least the point p . This, however, contradicts the fact that u is extreme in C . \square

Note that in the case of convex partitions, the general bound on the size of the resulting combination graph from Corollary 2 is insufficient to prove Theorem 2 because $\mathcal{U} = \mathcal{CP}_P$ alone can be of size $\Omega(2^n)$. The following lemma is therefore really needed.

Lemma 8. *For any point set P of size n and any subset \mathcal{U} of \mathcal{CP}_P , the size of $\Gamma_{\mathcal{U}}^{\text{pt}}$ is at most $O(2^n n^3)$.*

Proof. We prove that the number of labeled edges in $\Gamma_{\mathcal{U}}^{\text{pt}}$ can be bounded by $O(2^n n^3)$ from above, which then implies the lemma.

Fix $p_l, p_r \in P$ with $p_l \preceq p_r$, and let $P_l^r := \{p_l, p_{l+1}, \dots, p_{r-1}, p_r\} \subseteq P$ be the set of points between p_l and p_r . Observe that there are at most 2^{r-l+1} convex parts $u \in \mathcal{CP}_P$ for which $p_l = \text{lft}(u)$ and $p_r = \text{rgt}(u)$ holds. Let us also fix such a convex part u . Next, we will give a bound on the number of edges in $\Gamma_{\mathcal{U}}^{\text{pt}}$ with label u . This bound will only depend on the indices l and r .

Suppose that $[C] \xrightarrow{u} [C']$ is an edge in $\Gamma_{\mathcal{U}}^{\text{pt}}$ and assume further that $C \xrightarrow{u} C'$ holds. Then, we easily see that $\overline{\text{low}}(u) = \text{pts}(C) \cap P_l^r$, where $\overline{\text{low}}(u) := \text{low}(u) \setminus \text{pts}(u)$. This means that for each point $p \in P_l^r$ it is determined by u whether $p \in \text{pts}(C)$ holds or not. For a point $p \in P \setminus P_l^r$ there are at most two choices, either $p \in \text{pts}(C)$ or $p \notin \text{pts}(C)$. It follows that there are at most $2^{n-(r-l+1)}n$ many vertices $[C]$ with an outgoing edge that is labeled by u . As usual, the additional factor n comes from the special marking.

The total number of labeled edges in $\Gamma_{\mathcal{U}}^{\text{pt}}$ can therefore be bounded by

$$\sum_{l=1}^n \sum_{r=l}^n 2^{r-l+1} \cdot 2^{n-(r-l+1)} n = \sum_{l=1}^n \sum_{r=l}^n 2^n n = O(2^n n^3).$$

\square

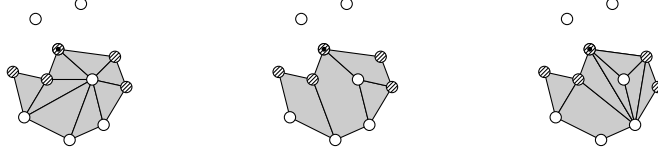


Figure 6: Three elements of $\mathfrak{C}^{\text{sd}}(\mathcal{CF}_P)$ that are considered equivalent.

4.3 Subdivisions

Let $\mathcal{U} \subseteq \mathcal{CF}_P$, which means in particular that the *shape* of each unit u is defined. We define the set $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ which contains all combinations C of \mathcal{U} for which the following holds. There exists an x -monotone polygonal chain, denoted by $\text{chn}(C)$, which starts in p_1 , ends in p_n , has only points from P as vertices, and satisfies the following with regard to C . The shapes of all u in C form a subdivision of the region between $\text{chn}(C)$ and the lower convex hull of P , by which we mean that the shapes are pairwise interior-disjoint and each point of the plane in the interior of that region is contained in the shape of at least one element u of C .

As depicted in Figure 6, we describe such a combination C by giving the vertex points of $\text{chn}(C)$ the color \otimes , by giving all other points the color \circ , and by adding the usual marking. Guided by this description, we put $C \sim C'$ if and only if

- $\text{chn}(C) = \text{chn}(C')$,
- $\text{lft}(\text{rex}(C)) = \text{lft}(\text{rex}(C'))$.

Lemma 9. *Let $\mathcal{U} \subseteq \mathcal{CF}_P$ with $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ serializable. Then, the equivalence relation \sim on $\mathfrak{C}^{\text{sd}}(\mathcal{U})$, as defined above, is coherent.*

Proof. Let $C_1, C_2 \in \mathfrak{C}^{\text{sd}}(\mathcal{U})$ be non-empty (otherwise, the proof is trivial) with $C_1 \sim C_2$ and assume that $C_1 \xrightarrow{u} C'_1$ holds for $C'_1 \in \mathfrak{C}^{\text{sd}}(\mathcal{U})$ and $u \in \mathcal{U}$. Consider $C'_2 := C_2 \cup \{u\}$. We show $C'_2 \in \mathfrak{C}^{\text{sd}}(\mathcal{U})$, $C'_1 \sim C'_2$ and $C_2 \xrightarrow{u} C'_2$, which implies the lemma.

From the definition of $\mathfrak{C}^{\text{sd}}(\mathcal{U})$, it is immediate that also $C'_2 \in \mathfrak{C}^{\text{sd}}(\mathcal{U})$ and $\text{chn}(C'_2) = \text{chn}(C'_1)$. Since u is extreme in C'_1 , it is also extreme in C'_2 . Using the usual argument involving the marking we see that $\text{rex}(C'_2) = u$, it follows that $C'_1 \sim C'_2$ and $C_2 \xrightarrow{u} C'_2$. \square

Let $\mathfrak{T} \subseteq (\mathfrak{C}^{\text{sd}}(\mathcal{U})/\sim)$ be the set that contains $[C]$ if and only if $\text{chn}(C)$ is equal to the upper convex hull of P . Observe that $\bigcup \mathfrak{T} = \mathfrak{C}^{\text{sd}}(\mathcal{U})$. If \mathcal{U} is such that $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ is serializable, then $(\mathcal{U}, \mathfrak{C}^{\text{sd}}(\mathcal{U}), \sim, \mathfrak{T})$ is a combination problem and we denote by $\Gamma_{\mathcal{U}}^{\text{sd}}$ the corresponding combination graph.

Corollary 3. *If $\mathcal{U} \subseteq \mathcal{CF}_P$ is such that $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ is serializable, then $\Gamma_{\mathcal{U}}^{\text{sd}}$ represents $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ and the size of $\Gamma_{\mathcal{U}}^{\text{sd}}$ is at most $O(|(\mathfrak{C}^{\text{sd}}(\mathcal{U})/\sim)| \cdot |\mathcal{U}|)$.*

Theorems 4 and 5 now follow from Corollary 3 and the following two lemmas.

Lemma 10. *For any point set P and any subset \mathcal{U} of \mathcal{CF}_P , it holds that $\mathfrak{C}^{\text{sd}}(\mathcal{U})$ is serializable.*

Proof. The proof for the existence of right-most extreme elements is again analogous to Lemma 4. Moreover, it is not hard to see that $C' := C \setminus \{u\}$, where $u = \text{rex}(C)$, is also an element of $\mathfrak{C}^{\text{sd}}(\mathcal{U})$. Simply observe that the upper convex hull of the shape of u must be contained in $\text{chn}(C)$, which means that $\text{chn}(C')$ is obtained from $\text{chn}(C)$ by replacing the upper hull of u with its lower hull. \square

Similar to the previous subsection, the next lemma is not implied by the general bound from Corollary 3.

Lemma 11. *For any point set P of size n and any subset \mathcal{U} of \mathcal{CF}_P , the size of $\Gamma_{\mathcal{U}}^{\text{sd}}$ is at most $O(2^n n^3)$.*

Proof. Analogous to the proof of Lemma 8. \square

5 Representations of Combination Graphs

In the preceding sections, vertices of a combination graph Γ were always treated as equivalence classes over a set of combinations. Of course, this is a very inefficient way to represent them in an actual algorithm. However, in all cases we have seen how to describe these equivalence classes by an assignment of a constant number of colors to the points in P and an index to the point with the special marking. It is thus easy to encode vertices of Γ with a linear number of bits for the coloring and a logarithmic number of bits for the index.

As for the construction of Γ , this is most easily done bottom-up. That is, we start with the source \perp , we enumerate all outgoing edges and add the corresponding new vertices to our representation of Γ . We continue this process, that is, pick a vertex and enumerate all outgoing edges, until we have done so for all vertices in Γ .

In all cases except for convex partitions and convex subdivisions, the enumeration of outgoing edges for a given vertex can be done in the most wasteful way while still staying within the required time bounds. For example, in the case of Theorem 1, for every vertex in $\Gamma_{\mathcal{S}_P}^{\text{cf}}$ we can simply enumerate the whole set \mathcal{S}_P and check for each segment whether it corresponds to an outgoing edge in linear time. In the case of Theorems 2 and 4 we have to be more careful since we do not have the time to enumerate the whole set \mathcal{CP}_P , say, for every vertex in $\Gamma_{\mathcal{CP}_P}^{\text{pt}}$. A simple solution is to enumerate the set \mathcal{CP}_P only once at the beginning and to find for each enumerated $u \in \mathcal{CP}_P$ all vertices in $\Gamma_{\mathcal{CP}_P}^{\text{pt}}$ for which u corresponds to an outgoing edge. The running time of this approach can be bounded by using the same arguments as in the proof of Lemma 8.

6 Enumeration with Polynomial-Time Delay

In this section we sketch an additional trick⁴, which allows us to enumerate the sets of crossing-free convex partitions and perfect matchings with polynomial-time delay. That is, we give algorithms which output the elements of $\mathcal{CP}(P)$ or $\mathcal{M}(P)$ in some order and without repetitions, and such that the time we have to wait for any new output is not larger than a polynomial.

⁴This trick was suggested to the author by Emo Welzl.

As already discussed in Section 2, from Theorems 2 and 3 we get enumeration algorithms for these two sets that run in time $O^*(\text{cp}(P))$ and $O^*(\text{pm}(P))$, respectively. However, while the time delay *between* any two outputs is bounded by a polynomial, the time delay *before* the first output is exponential. Precisely, there is a preprocessing phase that might take time $\Theta^*(2^n)$ during which we construct representations of the respective combination graphs $\Gamma_{\mathcal{CP}(P)}^{\text{pt}}$ and $\Gamma_{\mathcal{M}(P)}^{\text{pt}}$ and during which we do not produce any outputs. The trick is to hide this preprocessing phase by outputting $\Theta^*(2^n)$ objects obtained by other means.

Theorem 8. *The sets $\mathcal{CP}(P)$ and $\mathcal{M}(P)$ can be enumerated such that the time delay for any output is bounded by a polynomial in n .*

Proof. We begin by defining a sufficiently large subset $\mathcal{EM}(P)$ of $\mathcal{CP}(P)$ and $\mathcal{M}(P)$. Elements of this set are called *easy perfect matchings*, and they are constructed recursively. If P is the empty set, then $\mathcal{EM}(P) = \mathcal{M}(P)$. Otherwise, let p_1 be the left-most point and let p_i be any other point. Let P_i be the set of points that are to the left of the directed line through p_1 and p_i , and let P'_i be the set of points that are to the right of that line. The set $\mathcal{EM}(P)$ contains all perfect matchings that, for any choice of i , are composed of the edge p_1p_i and two easy perfect matchings on P_i and P'_i , respectively.

It is clear that easy perfect matchings are crossing-free. Furthermore, note that efficient enumeration and recognition algorithms for the set $\mathcal{EM}(P)$ are easy to obtain from the definition. Lastly, the number of easy perfect matchings on even-sized point sets satisfies the Catalan recurrence, and thus $|\mathcal{EM}(P)| = \Theta(2^n/n^{3/2})$. Refer to [11] for more details, where these objects are used to prove a corresponding lower bound on the number of perfect matchings.

We now have everything that we need. In order to enumerate the set $\mathcal{M}(P)$, say, with polynomial-time delay, we start the construction of $\Gamma_{\mathcal{M}(P)}^{\text{pt}}$. During this preprocessing phase, we output elements of $\mathcal{EM}(P)$ in appropriate time intervals. Once we have an explicit representation of $\Gamma_{\mathcal{M}(P)}^{\text{pt}}$, we continue the enumeration by outputting arbitrary elements of $\mathcal{M}(P)$. Of course, whenever we have a new potential output, we have to check first whether it is an easy perfect matching, which means that it has been output before. If that is the case, then we simply discard it. One final caveat now is that there might be a long period where we have to discard all potential outputs, which might again lead to a delay that is no longer polynomially bounded. However, this is easily fixed for example by only using up half of the set $\mathcal{EM}(P)$ during the preprocessing phase, and by using the other half as a substitute for every other discarded output during the second phase. \square

7 Spanning Trees and Spanning Cycles

In this section we show that it is possible to construct non-trivial combination graphs for the sets of crossing-free spanning trees and spanning cycles.

Spanning trees and spanning cycles are harder to deal with than anything that we have encountered before. The reason is that these graphs have properties which hold globally. For example, the construction of $\Gamma_{\mathcal{ST}(P)}^{\text{pt}}$ in Section 4.2 can be adapted in such a way that source-sink paths correspond to 2-regular (instead of 1-regular) crossing-free geometric graphs. We simply would have to keep track of the degree of each vertex (whether it is currently 0, 1, or 2, which means we

would need three instead of two colors) and in the end require that every vertex has degree 2. However, if we want that source-sink paths correspond only to crossing-free spanning cycles, then we also need that each such path corresponds to a connected geometric graph. Being connected is such a property that holds globally, and there seems to be no obvious and efficient way to deal with it. To get rid of this problem, at least in the case of spanning trees and spanning cycles, we next state an auxiliary lemma. It will allow us to translate connectivity into simpler features which can be enforced on a local level.

Let G be a directed multigraph⁵ and let v be a vertex in G . G is *root-oriented towards v* if all vertices in G have exactly one outgoing edge, except for v , which has no outgoing edges. If G is root-oriented towards v , then v is called the *root* of G . Observe that being root-oriented implies that G has exactly $n - 1$ edges, where n is the number of vertices in G . It does however not imply that G is connected or, in other words, a tree. The reason is that there might be a connected component with a directed cycle. Such components are always disconnected from the root.

A *plane drawing* of G is a drawing which maps all vertices of G to distinct points in the plane and which draws all edges as simple curves such that no two edges intersect except possibly in a common endpoint. Given two respective plane drawings of directed multigraphs G_1 and G_2 , we say the drawings are *disjoint* if they do not use any common points in the plane. Moreover, the drawings are *entangled* if for each cycle in either drawing, both its interior and exterior contain a point used by the respective other drawing. Finally, for fixed vertices v_1 in G_1 and v_2 in G_2 we say the two drawings are *tangent* in v_1 and v_2 if the points corresponding to v_1 and v_2 can be connected by an additional curve without intersecting any points already used in either drawing.

Lemma 12. *Let G_1 and G_2 be finite, directed multigraphs that are root-oriented towards v_1 and v_2 , respectively. Then, there exist plane drawings of G_1 and G_2 that are disjoint, entangled and tangent in v_1 and v_2 if and only if both G_1 and G_2 are trees.*

Proof. If G_1 and G_2 are trees then the desired drawings clearly exist. To prove the other direction of the lemma, it suffices to show that both G_1 and G_2 do not contain any cycles.

Let us thus fix plane drawings of G_1 and G_2 with the desired properties and assume that G_1 contains a cycle C . Without loss of generality, the root v_2 of G_2 is contained in the exterior of C . Since the drawings of G_1 and G_2 are tangent in v_1 and v_2 , also the root v_1 of G_1 is contained in the exterior of C . Now, from all cycles of either G_1 or G_2 that are contained in the interior of C , let us select a minimal cycle C' . Minimal means that C' does itself not contain any other cycles in its interior. Such a cycle exists since G_1 and G_2 are finite. We assume that C' again belongs to G_1 , the other case being analogous. Since the drawings of G_1 and G_2 are entangled, we get a vertex of G_2 in the interior of C' . Starting from this vertex we now follow directed edges in G_2 . Since the drawings of G_1 and G_2 are disjoint, we never leave the interior of C' , which in particular means that we never reach v_2 . However, since G_2 only has a finite number of vertices and all except for v_2 have an outgoing edge, we are bound

⁵To avoid confusion, let us stress that we introduce completely new entities here. That is, G is neither a geometric graph nor a combination graph.

to get into a cycle eventually. Clearly, this new cycle of G_2 is still contained in the interior of C' , in contradiction to minimality. \square

We make two adaptations to the abstract framework from Section 3. Note that all definitions and lemmas from that section extend naturally to the following setting.

A set of units \mathcal{U} is no longer understood as a simple subset of \mathcal{G}_P^* . Firstly, in this section we restrict units to be segments from the set \mathcal{S}_P . Secondly, a unit can have additional information attached to it. As an example, \mathcal{U} could be defined as the set of directed segments. That is, each u in \mathcal{U} would correspond to an element of \mathcal{S}_P , but it would also have a direction. In particular, this means that multiple elements of \mathcal{U} can correspond to the same geometric graph.

Moreover, a combination C of \mathcal{U} is still understood as a subset of \mathcal{U} . However, we do not allow the same geometric graph to appear twice in C . That is, in the above example, the elements of C must be pairwise distinct as segments. It is not sufficient if only their directions differ.

We conclude by giving some definitions and conventions that will be used in the following two subsections. We refer to Figure 7 for illustrations.

We assume there are unique points $\hat{p}, \check{p} \in P$ with largest and smallest y -coordinates, respectively. The horizontal line through \check{p} is called the *bottom*.

For every crossing-free combination C of some set of units \mathcal{U} we define a set of faces as follows. From the endpoints of each segment u in C we draw vertical rays (called *borders*) downwards until we hit either the bottom or the relative interior of another segment in C . Then, a *face* in C is a maximal connected region in the plane. There is one unbounded region above the bottom, which is called the *infinite face*. The unbounded region below the bottom is not a face and will be ignored. Furthermore, we say that two faces in C are *adjacent* if they share a (vertical) border.

Borders are always directed either left-to-right or right-to-left. In a combination C , the *out-degree* of a face is the number of borders directed away from that face. We further say that a point p *exposes a drain to the left* if the border below p is directed left-to-right and the region directly to the left of that border belongs to the infinite face. An analogous definition is given for *exposing a drain to the right*. If a point exposes a drain either to the left or to the right, we simply say that it *exposes a drain*.

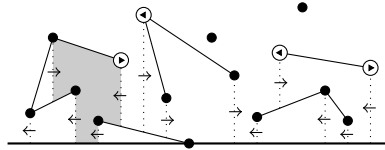


Figure 7: The thick line is the bottom. Dotted lines are borders. The shaded region is a face with out-degree 1. Points which expose a drain to the left are marked with \odot . Points which expose a drain to the right are marked with \odot .

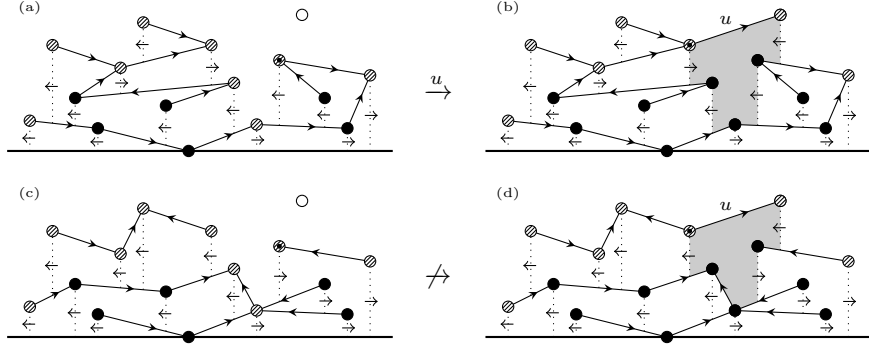


Figure 8: The elements of $\mathfrak{C}_P^{\text{st}}$ in (a) and (c) cannot be considered equivalent.

7.1 Spanning Trees

We define a very special set $\mathcal{S}_P^{\text{st}}$ of units. Each u in $\mathcal{S}_P^{\text{st}}$ is a segment from the set \mathcal{S}_P with a direction. Additionally, below either endpoint of u a border might be attached that is directed either left-to-right or right-to-left.

We also define the set $\mathfrak{C}_P^{\text{st}}$, which contains all crossing-free combinations C of $\mathcal{S}_P^{\text{st}}$ with the following additional properties.⁶ In what follows, the *out-degree* of a point $p \in P$ in C denotes the number of segments in C that have p as an endpoint and are directed away from p .

- The point \hat{p} has out-degree 0 in C .
- Every point $p \in P$ has out-degree at most 1 in C .
- Every point $p \in \text{low}(C)$ has out-degree 1 in C .
- Every finite face in C has out-degree 1.

Examples can be seen in Figure 8 (a), (b) and (c). The combination in (d) violates the last three properties.

Note that all of the above properties are maintained when removing the right-most extreme element from a combination from the set $\mathfrak{C}_P^{\text{st}}$. We thus get the following lemma.

Lemma 13. *For any point set P , $\mathfrak{C}_P^{\text{st}}$ is serializable.*

We reuse the three colors $\circ, \emptyset, \bullet$ and the special marking from Section 4.1 with their original meaning to describe elements of $\mathfrak{C}_P^{\text{st}}$, as already depicted in Figure 8. However, that same figure illustrates that we cannot reuse the old equivalence relation because it is not coherent. There are three problems we have to deal with. Firstly, the out-degree of a point can become larger than 1. Secondly, a point with out-degree 0 can disappear in the lower shadow of a segment. Thirdly, a finite face with out-degree not equal to 1 can be created.

⁶There is one technicality we gloss over, which however can be made precise: We also require that for each $p \in \text{pts}(C)$ only the first segment u in C with p as an endpoint has a border attached below p , where first refers to the usual order induced by serializability of $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$. In this way, for each picture in Figure 8 (a), (b) and (c), there exists a unique corresponding object in $\mathfrak{C}_P^{\text{st}}$.

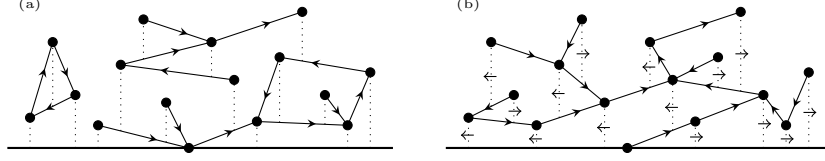


Figure 9: Illustrations for the proof of Lemma 15.

To make the equivalence relation coherent, it suffices to partition the points with color \otimes into six smaller categories. That is, we have to replace the color \otimes with 6 new colors, resulting in a total of 8 colors, and then consider two combinations equivalent if they agree in that new coloring and also in the special marking. We will not define the colors explicitly here, but only explain what information we have to keep track of.

For each point with color \otimes we keep track of its out-degree, that is, whether it is currently 0 or 1. This allows us to avoid the first two problems mentioned earlier. Furthermore, for each point with color \otimes we keep track of whether and where it exposes a drain. This allows us to avoid the third problem because whenever a new segment u is added to a combination C , a new finite face is created below u , and the out-degree of that face is determined by the number of exposed drains in the lower shadow and at the endpoints of u . Indeed, observe that borders corresponding to exposed drains in the lower shadow of u become out-borders of the new finite face. Also, an exposed drain at the left endpoint, say, of u becomes an out-border of the new face if and only if it is exposed to the right. This is why we also have to know the side a drain is exposed to.

We define an equivalence relation \sim on $\mathfrak{C}_P^{\text{st}}$ based on the $2 + 2 \cdot 3 = 8$ colors from the preceding discussion, and with the usual marking. The above intuition can be made precise, and the following can be proved.

Lemma 14. *The equivalence relation \sim on $\mathfrak{C}_P^{\text{st}}$, as defined above, is coherent.*

We define the set $\mathfrak{T} \subseteq (\mathfrak{C}_P^{\text{st}}/\sim)$ which contains $[C]$ if and only if every point (except for \hat{p}) has out-degree 1 in C and the infinite face has out-degree 0 (equivalently, there are no exposed drains in C). Let Γ_P^{st} be the combination graph corresponding to the combination problem $(\mathcal{S}_P^{\text{st}}, \mathfrak{C}_P^{\text{st}}, \sim, \mathfrak{T})$. Theorem 6 with $c = 8$ now follows from Lemma 2 and from the following insight.

Lemma 15. *There is a natural bijection between the sets $\bigcup \mathfrak{T}$ and $ST(P)$ in the following sense. For any combination C in $\bigcup \mathfrak{T}$, building the geometric graph on P with edges that correspond to the segments in C yields the corresponding crossing-free spanning tree in $ST(P)$.*

Proof. Any C in $\bigcup \mathfrak{T}$ induces two directed multigraphs G_1 and G_2 with corresponding plane drawings, as follows.

G_1 is the graph with vertex set P and edges that correspond to the directed segments in C . By definition of $\mathfrak{C}_P^{\text{st}}$ and \mathfrak{T} , we at least know that G_1 is root-oriented towards \hat{p} , as exemplified in Figure 9 (a).

For the vertices of G_2 we choose one arbitrary point in the interior of each face in C . Two vertices in G_2 are connected if their corresponding faces in C are adjacent. The direction of that edge is chosen in accordance with the

direction of the corresponding border in C . Again by definition of $\mathfrak{C}_P^{\text{st}}$ and \mathfrak{T} , G_2 is root-oriented towards the vertex corresponding to the infinite face in C , as exemplified in Figure 9 (b).

It is clear that the drawings of G_1 and G_2 can be chosen such that they are disjoint, entangled, and tangent in \hat{p} and the infinite face. The proof is concluded by applying Lemma 12 and by observing that any spanning tree on P can be root-oriented towards \hat{p} in a unique way. \square

One can prove that $c < 8$ by adapting the arguments from Section 4.1. With some more work, we get $c < 7.04313$, as will be shown in Section 8.

7.2 Spanning Cycles

We define a slightly different set of units $\mathcal{S}_P^{\text{sc}}$. In the same way as in the previous subsection, below the endpoints of any segment u in $\mathcal{S}_P^{\text{sc}}$ directed borders can be attached. Here, however, the segment u itself does not have a direction.

We also define the set $\mathfrak{C}_P^{\text{sc}}$, which contains all crossing-free combinations C of $\mathcal{S}_P^{\text{sc}}$ with the following additional properties. In what follows, the *degree* of a point $p \in P$ in C stands for the number of segments in C that have p as an endpoint. Also, if the size of C is n , the *last finite face* in C is defined as the face directly below the right-most extreme segment in C . All other finite faces are called *normal*.

- If $|C| = n$, the last finite face in C has out-degree 0.
- Every point $p \in P$ has degree at most 2 in C .
- Every point $p \in \text{low}(C)$ has degree 2 in C .
- Every normal finite face in C has out-degree 1.

Note again that the above properties are maintained when removing the right-most extreme element from a combination from the set $\mathfrak{C}_P^{\text{sc}}$.

Lemma 16. *For any point set P , $\mathfrak{C}_P^{\text{sc}}$ is serializable.*

It is clear that we have to keep track of the degrees of all points. For one last time, we change the meaning of the colors \circ , \otimes , \bullet and use them to identify points of degree 0, 1, and 2, respectively.

Unsurprisingly, and similar to the previous subsection, an equivalence relation based only on these three colors and the usual marking is not coherent, as illustrated in Figure 10. The only problem, however, is that finite faces which do not have the right out-degree can be created. A by now routine proof shows that these three colors are already sufficient to avoid crossings.

To avoid finite faces with wrong out-degrees, we split \otimes into 3 subcolors, and we split \bullet into 2 subcolors, giving us a total of 6 different colors which are then used to define an equivalence relation \sim on $\mathfrak{C}_P^{\text{sc}}$. For each point with color \otimes we keep track of whether it exposes a drain. If it does, then we also keep track of whether it is to the left or to the right. Remember, this extra information is relevant if and only if the point in question is one of the endpoints of a new segment. For a point with color \bullet we only keep track of whether it exposes a drain or not. The reason why this is sufficient is that such a point has, by definition, degree 2 already and cannot be an endpoint of a new segment.

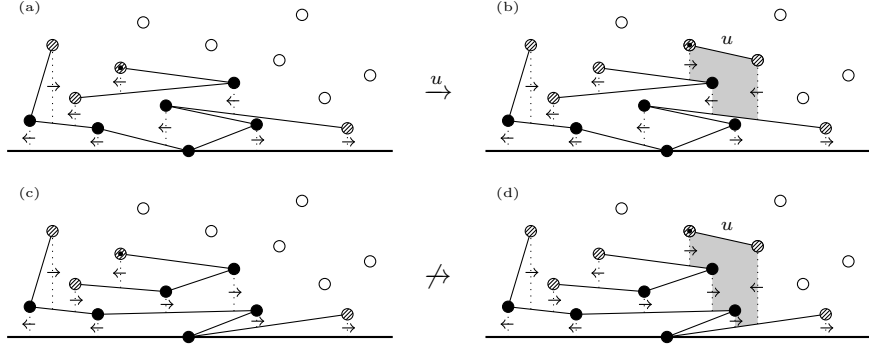


Figure 10: The elements of $\mathfrak{C}_P^{\text{sc}}$ in (a) and (c) cannot be considered equivalent.

Lemma 17. *The equivalence relation \sim on $\mathfrak{C}_P^{\text{sc}}$, as defined above, is coherent.*

We define the set $\mathfrak{T} \subseteq (\mathfrak{C}_P^{\text{sc}}/\sim)$ which contains $[C]$ if and only if every point has degree 2 in C and the infinite face has out-degree 0. Let Γ_P^{sc} be the combination graph corresponding to $(\mathcal{S}_P^{\text{sc}}, \mathfrak{C}_P^{\text{sc}}, \sim, \mathfrak{T})$. Theorem 7 with $c = 6$ now follows from Lemma 2 and from the following insight. The better bound $c < 5.61804$ will be proved in Section 8.

Lemma 18. *There is a natural bijection between the sets $\bigcup \mathfrak{T}$ and $SC(P)$ in the following sense. For any combination C in $\bigcup \mathfrak{T}$, building the geometric graph on P with edges that correspond to the segments in C yields the corresponding crossing-free spanning cycle in $SC(P)$.*

Proof. For any C in $\bigcup \mathfrak{T}$ we know that each point is of degree 2, which means that C is a set of disjoint cycles, as exemplified in Figure 11 (a). Similar to the proof of Lemma 15, C induces two directed multigraphs G_E and G_O .

The vertex set of G_E is the set of faces in C contained in an even number of cycles. The vertex set of G_O is the set of faces in C contained in an odd number of cycles. In both G_E and G_O , two vertices are connected by an edge if the corresponding faces in C are adjacent. The direction of each edge reflects the direction of the corresponding border in C . By definition of $\mathfrak{C}_P^{\text{sc}}$ and \mathfrak{T} , G_E and G_O are root-oriented towards the infinite face and the last finite face in C , respectively, as exemplified in Figure 11 (b).

Clearly, there exist plane drawings of G_E and G_O which are disjoint, entangled, and also tangent in the respective roots. Applying Lemma 12 hence concludes the proof. \square

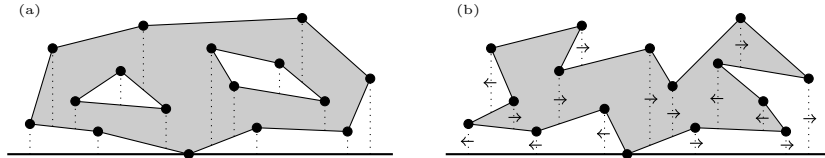


Figure 11: White faces are vertices in G_E . Shaded faces are vertices in G_O .

8 Bounding the Size of Combination Graphs

In this final section, we show how to prove the bounds on the constants c in Theorems 1, 6 and 7. Recall that vertices of combination graphs can be interpreted as colorings of P with a finite number of colors. The following proofs make use of the fact that certain patterns cannot occur in these colorings because of geometric constraints. Additionally, in the case of spanning trees and spanning cycles, we can improve the bounds further by identifying and discarding vertices from which the sink \top can no longer be reached.

8.1 All geometric graphs

Recall the definition of the set $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ and the corresponding equivalence relation \sim from Section 4.1. We are left to prove the following lemma.

Lemma 5. *For any set P of n points, the relation \sim partitions $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$ into at most $O(\alpha^n n)$ equivalence classes, that is, $|(\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)/\sim)| = O(\alpha^n n)$, where $\alpha \lesssim 2.83929$.*

Proof. We may encode equivalence classes $[C]$ in $(\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)/\sim)$ by a string s_C of length n over the alphabet $\{\bullet, \otimes, \circ\}$ and by an index k_C . The i -th entry in s_C is \bullet if $p_i \in \text{low}(C)$, it is \otimes if $p_i \in \text{pts}(C) \setminus \text{low}(C)$, and it is \circ otherwise. The index k_C is equal to the number i which satisfies $p_i = \text{lft}(\text{rex}(C))$. From this encoding we immediately get a bound of $O(3^n n)$ on the size of $(\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)/\sim)$.

Our proof strategy now is as follows. We ignore the indices k_C and give an upper bound of $O(\alpha^n)$ on the number of strings s_C which correspond to a combination C in $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$. From this, the desired upper bound easily follows after adding an additional factor n . We do so by defining an injective function from a set A to a set B . Set A contains all strings s_C . Set B contains all strings from $\{\bullet, \otimes, \circ\}^n$ which do not contain any subsequences of the form $(\bullet, \circ, \dots, \circ, \bullet)$, that is, one or more consecutive symbols \circ enclosed by two symbols \bullet . Such subsequences are called *forbidden* henceforth. The bound on $|A|$ then follows from $|B| = \Theta(\alpha^n)$, an elementary counting problem whose proof we omit.

Let us define such an injective function $f: A \rightarrow B$. For any $a \in A$ we construct $f(a)$ by the following process. We iterate over a from left to right, and whenever we find a forbidden subsequence $(\bullet, \circ, \dots, \circ, \bullet)$ we replace it by a sequence $(\otimes, \bullet, \dots, \bullet, \otimes)$ of the same length. For example, if $a = (\bullet, \circ, \bullet, \bullet, \bullet, \circ, \bullet)$ then $f(a) = (\otimes, \bullet, \otimes, \bullet, \otimes, \bullet, \otimes)$. However, if $a = (\bullet, \circ, \bullet, \circ, \bullet)$ then $f(a) = (\otimes, \bullet, \otimes, \circ, \bullet)$ because the second forbidden subsequence in a is no longer a forbidden subsequence after the first one has been replaced.

Clearly, f is a function $A \rightarrow B$. It only remains to prove injectivity. Towards a contradiction, assume thus that $a \neq a'$ satisfy $f(a) = f(a') =: b$. Let i be the smallest index with $a_i \neq a'_i$. We distinguish the three cases $b_i = \circ$, $b_i = \otimes$ and $b_i = \bullet$.

For $b_i = \circ$, observe that the function f never uses the symbol \circ to replace an entry in a or a' . Hence, $a_i = a'_i = \circ$, a contradiction.

For $b_i = \otimes$, we may assume without loss of generality that $a_i = \bullet$ and $a'_i = \otimes$. Furthermore, a_i must be either the first or last letter in a forbidden subsequence in a that is replaced under f . From minimality of i it follows that it is the first letter. Since the following argument generalizes to larger forbidden subsequences, we can now assume for reasons of simplicity that we

have $(a_i, a_{i+1}, a_{i+2}) = (\bullet, \circ, \bullet)$ and $(b_i, b_{i+1}, b_{i+2}) = (\otimes, \bullet, \otimes)$. There are two possibilities for the corresponding letters in a' .

- If $(a'_i, a'_{i+1}, a'_{i+2}) = (\otimes, \bullet, \otimes)$, then observe that the points p_i, p_{i+1}, p_{i+2} from the set P must form a left turn, that is, p_{i+1} is strictly below the segment with endpoints p_i and p_{i+2} . The reason is that $a' = s_{C'}$ corresponds to a combination C' in $\mathfrak{C}^{\text{cf}}(\mathcal{S}_P)$, and there must be a segment in C' which has p_{i+1} , but neither p_i nor p_{i+2} , in its lower shadow. On the other hand, $(a_i, a_{i+1}, a_{i+2}) = (\bullet, \circ, \bullet)$ implies that the points p_i, p_{i+1}, p_{i+2} form a right turn, that is, p_{i+1} is strictly above the segment with endpoints p_i and p_{i+2} . Again, this holds because also $a = s_C$ corresponds to a combination C , and there must be a segment in C which has both p_i and p_{i+2} , but not p_{i+1} , in its lower shadow. We have derived a contradiction because these two arrangements of p_i, p_{i+1} and p_{i+2} are mutually exclusive.
- If $(a'_i, \dots, a'_{i+3}) = (\otimes, \bullet, \bullet, \circ)$ and $(b_i, \dots, b_{i+3}) = (\otimes, \bullet, \otimes, \bullet)$, that is, index $i+2$ is the beginning of a forbidden subsequence in a' , then we further obtain $(a_i, \dots, a_{i+3}) = (\bullet, \circ, \bullet, \bullet)$. A contradiction can be derived in the same manner as in the previous case. The symbols in a' imply that the points p_i, p_{i+1} and p_{i+3} form a left turn because there exists a segment which has p_{i+1} and p_{i+2} , but neither p_i nor p_{i+3} , in its lower shadow. The symbols in a imply that p_i, p_{i+1} and p_{i+3} form a right turn because there exists a segment which has p_i, p_{i+2} and p_{i+3} , but not p_{i+1} , in its lower shadow.

For $b_i = \bullet$, we may assume that $a_i = \circ, a'_i = \bullet$, and that in a there exists a forbidden subsequence that starts before index i and ends after index i and that is replaced under f . Again for simplicity we assume $(a_{i-1}, a_i, a_{i+1}) = (\bullet, \circ, \bullet)$ and $(b_{i-1}, b_i, b_{i+1}) = (\otimes, \bullet, \otimes)$, longer sequences can be treated similarly. By minimality of i we get $a'_{i-1} = a_{i-1} = \bullet$. The only possible way for $a'_{i-1} = \bullet$ to be replaced by $b_{i-1} = \otimes$ under f is if a'_{i-1} is the last letter in a forbidden subsequence in a' . Therefore, $(a'_{i-3}, a'_{i-2}, a'_{i-1}) = (\bullet, \circ, \bullet)$ and $(b_{i-3}, b_{i-2}, b_{i-1}) = (\otimes, \bullet, \otimes)$, again without loss of generality. By minimality of i we further get $(a_{i-3}, a_{i-2}, a_{i-1}) = (\bullet, \circ, \bullet)$. In summary, we have derived $(a_{i-3}, \dots, a_{i+1}) = (\bullet, \circ, \bullet, \circ, \bullet)$ and $(b_{i-3}, \dots, b_{i+1}) = (\otimes, \bullet, \otimes, \bullet, \otimes)$. This is a contradiction because at most one of these two forbidden subsequences in a is replaced under f . \square

8.2 Spanning Trees

Recall the definition of Γ_P^{st} from Section 7.1. The following lemma is all that is left to complete the proof of Theorem 6.

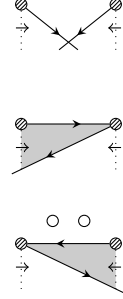
Lemma 19. *For any P , there exists a subgraph of Γ_P^{st} induced by a vertex subset $V \subseteq (\mathfrak{C}_P^{\text{st}}/\sim)$ satisfying $|V| = O(\alpha^n n)$, where $\alpha \lesssim 7.04313$, and such that V contains all vertices that appear on at least one $\perp\text{-}\top$ path in Γ_P^{st} . Moreover, given a vertex of Γ_P^{st} , we can decide in time $O(n)$ whether it belongs to V .*

Note that since we are able to recognize vertices from the set V efficiently, the smaller subgraph of Γ_P^{st} can also be constructed efficiently bottom-up, simply by discarding any encountered vertices that do not belong to V .

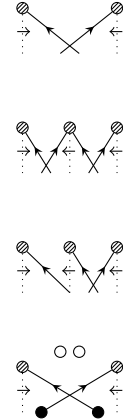
Proof. We begin by specifying the eight colors $\circledcirc, \circledast, \dots, \circledcirc$ that were introduced only informally in Section 7.1. Given a combination C in $\mathfrak{C}_P^{\text{st}}$, these colors are assigned to the points in P as follows. If $p \in \text{low}(C)$ then p has color \circledcirc , that is, $\circledcirc = \bullet$ in the original color scheme. Otherwise, and if additionally $p \notin \text{pts}(C)$, then p has color \circledcirc , that is, $\circledcirc = \circ$. As follows, all remaining points are assigned one of the colors $\circledcirc, \dots, \circledcirc$, which thus correspond to the original \circledcirc . Color \circledcirc means that p has out-degree 0 and exposes no drain. Color \circledcirc means that p has out-degree 0 and exposes a drain to the left. Color \circledcirc means that p has out-degree 0 and exposes a drain to the right. Color \circledcirc means that p has out-degree 1 and exposes no drain. Color \circledcirc means that p has out-degree 1 and exposes a drain to the left. Color \circledcirc means that p has out-degree 1 and exposes a drain to the right.

Similar to the proof of Lemma 5, let s_C be the string from $\{\circledcirc, \dots, \circledcirc\}^n$ that corresponds to a given combination C in $\mathfrak{C}_P^{\text{st}}$. The present lemma is a consequence of the following three observations, which all impose restrictions on certain patterns in s_C .

- For two consecutive points p_i and p_{i+1} , it cannot be that both p_i and p_{i+1} have out-degree 1 and, at the same time, p_i exposes a drain to the left and p_{i+1} exposes a drain to the right. As illustrated on the right, if that were the case then there would be either a crossing or a finite face of out-degree 0, both in contradiction with the definition of $\mathfrak{C}_P^{\text{st}}$. Hence, in s_C we will never observe the pattern $(\circledcirc, \circledcirc)$, that is, color \circledcirc directly followed by color \circledcirc . Additionally, if p_i and p_{i+k} (taking the role of p_{i+1}) are separated by any number of points of degree 0, the same argument still applies. Hence, we can further rule out the pattern $(\circledcirc, \circledcirc, \dots, \circledcirc, \circledcirc)$.



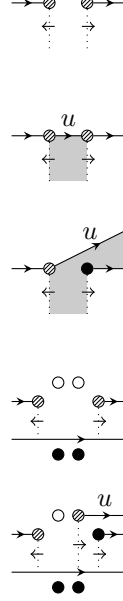
- Similarly, for two consecutive points p_i and p_{i+1} , it cannot be that both p_i and p_{i+1} have out-degree 0 and, at the same time, p_i exposes a drain to the left and p_{i+1} exposes a drain to the right. Since both points have out-degree 0 they cannot be connected by an edge, and thus we necessarily get a crossing, as illustrated on the right. Hence, the pattern $(\circledcirc, \circledcirc)$ cannot occur in s_C . Using the same argument we can further rule out the pattern $(\circledcirc, \circledcirc, \dots, \circledcirc, \circledcirc)$, since in such a configuration there will always be two consecutive points p_i and p_{i+1} of out-degree 0 such that p_i is the left endpoint of an edge and p_{i+1} is the right endpoint of another edge. Similar to the first observation, any additional points with degree 0 do not interfere with this argument. The same can be said for any points that are in the lower shadow of C . Hence, we can rule out the pattern $(\circledcirc, \circledcirc|\circledcirc, \dots, \circledcirc|\circledcirc, \circledcirc)$, where “|” indicates an arbitrary choice between several options.



Lastly, we describe a pattern which might actually occur in s_C . However, for any such occurrence we will prove that the corresponding vertex $[C]$ is not contained in any \perp - \top path in Γ_P^{st} . Therefore, the set V from the lemma may

safely be defined as the subset of all vertices of Γ_P^{st} that do not contain this final pattern.

- Let p_i and p_{i+1} be two consecutive points such that p_i exposes a drain to the right and p_{i+1} exposes a drain to the left. Then, clearly, the out-degree of the infinite face in C is different from 0 and thus there is no direct connection from $[C]$ to \top in Γ_P^{st} . Moreover, any new segment u that consumes the drain exposed by p_i (that is, the border below p_i becomes an out-border of the new face directly below u) must also consume the drain exposed by p_{i+1} , as illustrated on the right. However, the face below u having out-degree at least 2 contradicts the definition of $\mathfrak{C}_P^{\text{st}}$. Therefore, the two drains exposed by p_i and p_{i+1} can never be consumed and the sink \top can thus never be reached. Hence, we can safely discard any vertices with the pattern $(\textcircled{4}|\textcircled{7}, \textcircled{3}|\textcircled{6})$. Additionally, suppose that p_i and p_{i+k} (taking the role of p_{i+1}) are separated by any number of points of degree 0 or points in the lower shadow of C . Then, no matter how a new segment u is added, we again get a pair of drains facing each other and all points in between are either of degree 0 or in the lower shadow. Hence, we can further discard any vertices with the pattern $(\textcircled{4}|\textcircled{7}, \textcircled{3}|\textcircled{6}, \dots, \textcircled{3}|\textcircled{6}, \textcircled{3}|\textcircled{6})$.



Let A be the subset of $\{\textcircled{1}, \dots, \textcircled{6}\}^n$ containing only strings without subsequences belonging to the three patterns from the above observations. Then, it can be shown that $|A| = \Theta(\alpha^n)$ using standard techniques, and the lemma follows. \square

8.3 Spanning Cycles

Finally, recall the definition of Γ_P^{sc} from Section 7.2. The following lemma concludes the proof of Theorem 7.

Lemma 20. *For any P , there exists a subgraph of Γ_P^{sc} induced by a vertex subset $V \subseteq (\mathfrak{C}_P^{\text{sc}}/\sim)$ satisfying $|V| = O(\alpha^n n)$, where $\alpha \lesssim 5.61804$, and such that V contains all vertices that appear on at least one \perp - \top path in Γ_P^{sc} . Moreover, given a vertex of Γ_P^{sc} , we can decide in time $O(n)$ whether it belongs to V .*

Proof. We again start by giving specifications for the colors $\textcircled{1}, \textcircled{2}, \dots, \textcircled{6}$ that are assigned to the points in P for any given combination C in $\mathfrak{C}_P^{\text{sc}}$. Color $\textcircled{1}$ means that the corresponding point p has degree 2 and exposes no drain. Color $\textcircled{2}$ means that p has degree 2 and exposes a drain. Color $\textcircled{3}$ means that p has degree 1 and exposes no drain. Color $\textcircled{4}$ means that p has degree 1 and exposes a drain to the left. Color $\textcircled{5}$ means that p has degree 1 and exposes a drain to the right. Color $\textcircled{6}$ means that p has degree 0.

The following observation is similar to the third observation in the proof of Lemma 19. Let C be a combination in $\mathfrak{C}_P^{\text{sc}}$ and let p_i and p_{i+1} be two consecutive points. If p_i exposes a drain to the right and p_{i+1} exposes a drain to the left in

C , then C cannot be augmented in such a way that the infinite face has out-degree 0 without creating finite faces with out-degree larger than 1. In other words, the vertex $[C]$ does not appear on any $\perp\text{-}\top$ path in Γ_P^{sc} and may safely be discarded. The same holds for two not necessarily consecutive points p_i and p_{i+k} such that again p_i exposes a drain to the right and p_{i+k} exposes a drain to the left, and such that all points in between p_i and p_{i+k} have either degree 0 or 2. No matter how a new segment u is added to C , we will again end up with a combination with two exposed drains facing each other and only points of degree 0 or 2 in between.

The subset V of vertices of Γ_P^{sc} can thus be defined as follows. We simply exclude all vertices which contain the pattern $(\textcircled{\circ}, \textcircled{\circ}|\textcircled{\circ}|\textcircled{\circ}, \dots, \textcircled{\circ}|\textcircled{\circ}|\textcircled{\circ}, \textcircled{\circ})$. Let now A be the subset of $\{\textcircled{\circ}, \dots, \textcircled{\circ}\}^n$ containing only strings without subsequences belonging to the above pattern. Then, $|A| = \Theta(\alpha^n)$, and the lemma follows. \square

9 Acknowledgements

Most of the results presented in this paper were originally obtained in the author's master's thesis [25] in a, however, substantially less concise format. The author is deeply grateful and indebted to his then supervisor and now PhD advisor Emo Welzl. Special thanks go to Raimund Seidel for presenting his beautiful algorithm for counting triangulations at ETH Zürich in fall 2012, which undoubtedly marked the beginning of the success of that thesis.

References

- [1] AICHHOLZER, O. The path of a triangulation. In *Proc. 15th Symp. on Comp. Geometry* (1999), pp. 14–23.
- [2] AICHHOLZER, O., AURENHAMMER, F., HUEMER, C., AND VOGTENHUBER, B. Gray code enumeration of plane straight-line graphs. *Graphs and Combinatorics* 23, 5 (2007), 467–479.
- [3] AICHHOLZER, O., HACKL, T., HUEMER, C., HURTADO, F., KRASSER, H., AND VOGTENHUBER, B. On the number of plane geometric graphs. *Graphs and Combinatorics* 23, 1 (2007), 67–84.
- [4] ALVAREZ, V., BRINGMANN, K., CURTICAPEAN, R., AND RAY, S. Counting crossing-free structures. In *Proc. 28th Symp. on Comp. Geometry* (2012), pp. 61–68.
- [5] ALVAREZ, V., BRINGMANN, K., RAY, S., AND SEIDEL, R. Counting triangulations and other crossing-free structures approximately. *Computational Geometry* 48, 5 (2015), 386 – 397. Special Issue on the 25th Canadian Conference on Computational Geometry (CCCG).
- [6] ALVAREZ, V., AND SEIDEL, R. A simple aggregative algorithm for counting triangulations of planar point sets and related problems. In *Proc. 29th Symp. on Comp. Geometry* (2013), pp. 1–8.
- [7] AVIS, D., AND FUKUDA, K. Reverse search for enumeration. *Discrete Applied Math.* 65 (1996), 21–46.

- [8] BESPAMYATNIKH, S. An efficient algorithm for enumeration of triangulations. *Comput. Geom.* 23, 3 (2002), 271–279.
- [9] DUMITRESCU, A., SCHULZ, A., SHEFFER, A., AND TÓTH, C. D. Bounds on the maximum multiplicity of some common geometric graphs. In *Proc. 28th Symp. on Theoretical Aspects of Comp. Sci.* (2011), pp. 637–648.
- [10] FLAJOLET, P., AND NOY, M. Analytic combinatorics of non-crossing configurations. *Discrete Math.* 204 (1999), 203–229.
- [11] GARCÍA, A., NOY, M., AND TEJEL, J. Lower bounds on the number of crossing-free subgraphs of K_n . *Comput. Geom. Theory Appl.* 16, 4 (2000), 211–221.
- [12] HOFFMANN, M., SHARIR, M., SHEFFER, A., TÓTH, C. D., AND WELZL, E. Counting plane graphs: Flippability and its applications. In *Proc. 12th Symp. on Algs. and Data structs.* (2011), pp. 524–535.
- [13] HUEMER, C., AND DE MIER, A. Lower bounds on the maximum number of non-crossing acyclic graphs. *European Journal of Combinatorics* 48 (2015), 48 – 62. Selected Papers of EuroComb’13.
- [14] KARPINSKI, M., LINGAS, A., AND SLEDNEU, D. A QPTAS for the base of the number of crossing-free structures on a planar point set. In *ICALP’15* (2015), pp. 785–796.
- [15] KATOH, N., AND TANIGAWA, S. Fast enumeration algorithms for non-crossing geometric graphs. In *Proc. 24th Symp. on Comp. Geometry* (2008), pp. 328–337.
- [16] MARX, D., AND MILTZOW, T. Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems. *CoRR abs/1603.07340* (2016).
- [17] RAZEN, A., AND WELZL, E. Counting plane graphs with exponential speed-up. In *Rainbow of computer science*. Springer, 2011, pp. 36–46.
- [18] SHARIR, M., AND SHEFFER, A. Counting triangulations of planar point sets. *Electr. J. Comb.* 18, 1 (2011).
- [19] SHARIR, M., AND SHEFFER, A. Counting plane graphs: Cross-graph charging schemes. In *Proc. 20th Symp. on Graph Drawing* (2012), pp. 19–30.
- [20] SHARIR, M., SHEFFER, A., AND WELZL, E. On degrees in random triangulations of point sets. In *Proc. 26th Symp. on Comp. Geometry* (2010), pp. 297–306.
- [21] SHARIR, M., SHEFFER, A., AND WELZL, E. Counting plane graphs: Perfect matchings, spanning cycles, and kasteleyn’s technique. In *Proc. 28th Symp. on Comp. Geometry* (2012), pp. 189–198.
- [22] SHARIR, M., AND WELZL, E. On the number of crossing-free matchings, cycles, and partitions. *SIAM J. Comput.* 36, 3 (2006), 695–720.

- [23] SHEFFER, A. Numbers of plane graphs.
<http://www.cs.tau.ac.il/~sheffera/counting/PlaneGraphs.html>.
 Accessed: April 18th 2016.
- [24] TOUSSAINT, G. T. Movable separability of sets. In *Computational Geometry* (1985), North-Holland, pp. 335–375.
- [25] WETTSTEIN, M. Algorithms for counting crossing-free configurations. Master’s thesis, ETH Zürich, April 2013.
- [26] WETTSTEIN, M. Counting and enumerating crossing-free geometric graphs. In *Proc. 30th Symp. on Comp. Geometry* (2014), pp. 1–10.